TRANSMITTAL OF APPEAL BRIEF

Docket No.
36159/098001; P5944

In re Application of: David S. Allison

Application No.	Filing Date	Examiner	Group Art Unit
09/977,715-Conf. #3306	October 12, 2001	L. B. Zhen	2194

Invention: METHOD AND APPARATUS FOR COMMUNICATION TO THREADS OF CONTROL THROUGH STREAMS

TO THE COMMISSIONER OF PATENTS:

Transmitted herewith is the Appeal Brief in this application, with respect to the Notice of Appeal filed: July 18, 2008 .

The fee for filing this Appeal Brief is $ 540.00 .

[x] Large Entity [ ] Small Entity

[x] A petition for extension of time is also enclosed.

The fee for the extension of time is $ 130.00 .

[ ] A check in the amount of _____ is enclosed.

[ ] Charge the amount of the fee to Deposit Account No. 50-0591 .
This sheet is submitted in duplicate.

[x] Payment by credit card.

[x] The Director is hereby authorized to charge any additional fees that may be required or credit any overpayment to Deposit Account No. 50-0591 .

/Robert P. Lord/ Dated: October 20, 2008

Robert P. Lord
Attorney Reg. No. : 46,479
OSHA · LIANG LLP
909 Fannin Street, Suite 3500
Houston, Texas 77010
(713) 228-8600

Docket No.: 36159/098001; P5944
(PATENT)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:
David S. Allison

Confirmation No.: 3306

Application No.: 09/977,715

Art Unit: 2194

Filed: October 12, 2001

Examiner: L. B. Zhen

For: METHOD AND APPARATUS FOR COMMUNICATION TO THREADS OF CONTROL THROUGH STREAMS

MS Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPELLANTS' AMENDED BRIEF UNDER 37 CFR § 41.37

Pursuant to 37 CFR § 41.37, please consider the following Appellant's Brief in the referenced application currently before the Board of Patent Appeals and Interferences.

TABLE OF CONTENTS

TABLE OF AUTHORITIES

I. Real Party of Interest

The real party of interest for the referenced application is Sun Microsystems, Inc. An Assignment transferring all interest in the referenced application from the inventor to Sun Microsystems, Inc. was filed with the USPTO on October 12, 2001. The Assignment is recorded at Reel 012267, Frame 0435.

II. Related Appeals and Interferences

To the best of the knowledge of the Appellants and Appellants' legal representative, there are no other appeals or interferences that will directly affect, be affected by, or have a bearing on the decision of the Board of Patent Appeals and Interferences ("the Board") in this appeal.

III. Status of Claims

U.S. Application Serial No. 09/977,715 ("the '715 Application") was filed on October 12, 2001. As filed, the '715 Application included claims 1-22. In an amendment dated July 19, 2006, claims 1-22 were cancelled, and claims 23-36 were added. In an amendment dated July 9, 2007, claims 37 and 38 were added. In an amendment dated December 13, 2007, claims 24, 25, 31, and 32 were cancelled, and claims 39 and 40 were added. Accordingly, claims 23, 26-30, and 33-40 are pending in the '715 Application. Claims 23 and 30 are independent. The remaining claims depend, either directly or indirectly, from claims 23 and 30.

All the pending claims were finally rejected in an Office Action dated March 18, 2008 ("Final Rejection"). An Early Response to the Final Rejection (*i.e.* Request for Reconsideration)

was filed on May 12, 2008. The Final Rejection was sustained in an Advisory Action dated June 4, 2008 (hereinafter "Advisory").

Claims 23, 26-30, and 33-40 are on appeal.

IV. Status of Amendments

All of the amendments have been entered and considered by the Examiner. No amendments have been filed subsequent to the Final Rejection. The pending claims of record are presented in the Claims Appendix.

V. Summary of Claimed Subject Matter

Independent claim 23 relates to a method for communicating between threads. The method comprises: invoking a first thread; associating a first input stream and a first output stream with the first thread; invoking a second thread; associating a second input stream and a second output stream with the second thread; invoking a stream operator to write a first data value from the first thread to the second thread, wherein the stream operator connects the first output stream to the second input stream and sends the first data value from the first output stream to the second input stream; using the second thread to generate a second data value by performing an operation on the first data value; and invoking the stream operator to write the second data value from the second thread to the first thread, wherein the stream operator connects the second output stream to the first input stream and sends the second data value from the second output stream to the first input stream, wherein at least one selected from the group consisting of the first thread and the second thread manages an operating system process and comprises: a program counter; a stack; a state; and a register set. The method recited in independent claim 23 is described in at least Figures 3C and 10,

and on at least page 16, lines 10-20; page 20, lines 1-10; and page 21, lines 7-16 of the originally filed specification.

Independent claim 30 relates to a computer readable medium storing instructions for communicating between threads. The instructions comprising functionality to: invoke a first thread; associate a first input stream and a first output stream with the first thread; invoke a second thread; associate a second input stream and a second output stream with the second thread; invoke a stream operator to write a first data value from the first thread to the second thread, wherein the stream operator connects the first output stream to the second input stream and wherein the stream operator sends the first data value from the first output stream to the second input stream; use the second thread to generate a second data value by performing an operation on the first data value; and invoke the stream operator to write the second data value from the second thread to the first thread, wherein the stream operator connects the second output stream to the first input stream and writes the data value from the second output stream to the first input stream, wherein at least one selected from the group consisting of the first thread and the second thread manages an operating system process and comprises: a program counter; a stack; a state; and a register set. The computer readable medium recited in independent claim 30 is described in at least Figures 3C and 10, and on at least page 16, lines 10-20; page 20, lines 1-10; and page 21, lines 7-16 of the originally filed specification

VI. Grounds of Rejection to be Reviewed on Appeal

The present Appeal addresses the following grounds of rejection:

- Whether claims 23, 26-30, and 33-40 are unpatentable under 35 U.S.C. § 103(a) over U.S. Patent No. 6,131,183 (hereinafter "Tyler") in view of U.S. Patent No. 6,842,898 (hereinafter "Carlson").

VII. Argument

A. Claims 23, 26-30, and 33-40 are patentable over Tyler and Carlson

In this Appeal, Appellants argue that claims 23, 26-30, and 33-40 are patentable over Tyler and Carlson, whether viewed separately or in combination, for at least the reasons given below. For the purposes of this Appeal, claims 23, 26-30, and 33-40 stand or fall together. Independent claim 23 is representative of the group including claims 23, 26-30, and 33-40.

MPEP § 2143 states that "[t]he key to supporting any rejection under 35 U.S.C. 103 is the clear articulation of the reason(s) why the claimed invention would have been obvious." The Supreme Court in *KSR International Co. v. Teleflex Inc.*, 127 S. Ct. 1727, 82 USPQ2d 1385, 1395-97 (2007) noted that the analysis supporting a rejection under 35 U.S.C. § 103 should be made explicit. In the Final Rejection, the Examiner, while articulating the analysis used to reject the claims under 35 U.S.C. § 103, has described the various claimed elements taught and not taught by Tyler. *See,* Final Rejection at pages 2-5. Further, the Examiner has described the various claimed elements taught by Carlson, which are not taught by Tyler. *Id.* The Examiner then concludes by asserting that it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Tyler to include the teaching of Carlson. Id.

Using the above rationale, the Examiner "must articulate the following: (1) a finding that the prior art included each element claimed, although not necessarily in a single prior art

reference, with the only difference between the claimed invention and the prior art being the lack of actual combination of the elements in a single prior art reference; …" MPEP § 2143 (A). Applicant respectfully submits that the Examiner has failed to do so.

Further, if the Examiner does not produce a *prima facie* case, Applicant is under no obligation to submit evidence of non-obviousness. The initial evaluation of *prima facie* obviousness thus relieves both the Examiner and Applicant from evaluating evidence beyond the prior art and the evidence in the specification as filed until the art has been shown to suggest the claimed invention. See, MPEP § 2142.

Independent claim 23 explicitly requires, at least, (*i*) a first data value be written from a first thread to a second thread; (*ii*) said first data value be written to the second thread using the output stream of the first thread and the input stream of the second thread; (*iii*) a second data value be generated by the second thread based on the first value; (*iv*) said second data value be written from the second thread to the first thread; and (*v*) said second data value be written to the first thread using the output stream of the second thread and the input stream of the first thread. Independent claim 30 has similar requirements.

1. Tyler and Carlson fail to teach or suggest the second data value is written from the second thread to the first thread using the output stream of the second thread.

The Examiner contends that Tyler discloses writing a first data value from a first thread to a second thread, and then using the second thread to generate a second value from the first value. The Examiner also contends that Tyler discloses writing the second data value to the first thread using the output stream of the second thread and the input stream of the first thread. In other words,

the Examiner contends that Tyler discloses requirements (*iii*), (*iv*), and (*v*) of the independent claims. *See,* Final Rejection at pages 2-4. Applicant respectfully disagrees with the Examiner's contentions. Specifically, Applicant respectfully asserts the Examiner is mischaracterizing the prior art, which is wholly improper.

Tyler discloses a controller having a *stdin* stream, a *stdout* stream, a *childWrite* stream, and a *childRead* stream. Tyler further discloses a program having a *stdin* stream and a *stdout* stream. The *childWrite* stream and the *childRead* stream of the controller are attached to the *stdin* stream and the *stdout* stream, respectively, of the program. Further still, Tyler discloses an X-server with independent connections to both said controller and said program. *See,* Tyler at Figure 6; column 4, lines 65-68; and at column 5, lines 1-10.

Moreover, Tyler discloses a wakeup subroutine which sends a command (hereinafter "command A") from the controller to the program. In response to receiving command A, the program sends a command (hereinafter "command B") to the X-server to delete a dummy window. The wakeup subroutine then repeatedly polls the X-server to determine if the dummy window is still present. As a result of the polling, information (hereinafter "polling result") is returned to the wakeup subroutine. *See,* Tyler at Figure 6 and at column 6, lines 20-30.

a. Case A

In Case A, the Examiner equates the wakeup subroutine and command A, as disclosed by Tyler, with the first thread and the first value, respectively, as recited by the independent claims. *See,* Advisory at page 3. The Examiner also equates the program and command B, as disclosed by

Tyler, with the second thread and the second value, respectively, as recited by the independent claims. *See,* Final Rejection at page 3, lines 1, 2, and 12-15.

Accordingly, in order for the Examiner's contentions to hold true, Tyler must teach or suggest at least command B (*i.e.,* the second value) is written to the wakeup subroutine (*i.e.,* first thread). However, as discussed above, Tyler actually discloses command B is written to the X-server, ***not*** to the wakeup subroutine. Tyler does not even contemplate writing command B to the wakeup subroutine, as this would effectively prevent command B from reaching the X-server and trigger the purging of a dummy window. Accordingly, as Tyler contradicts requirement (*iv*) of the independent claims, Tyler cannot satisfy requirement (*iv*) of the independent claims.

In addition, in order for the Examiner's contentions to hold true, Tyler must also teach or suggest command B (*i.e.,* the second value) is written using the *stdout* stream of the program (*i.e.,* output stream of the second thread). As discussed above, Tyler discloses command B is written to the X-server. However, Tyler also discloses the *stdout* stream of the program is connected to the controller, ***not*** to the X-server. Accordingly, because Tyler discloses command B is written to the X-server, Tyler effectively discloses command B is written ***without*** using the *stdout* stream of the program. This squarely contradicts requirement (*v*) of the independent claims. Accordingly, Tyler also cannot satisfy requirement (*v*) of the independent claims.

b. Case B

Case B is essentially the same as Case A. However, in Case B, the Examiner equates the polling result, as disclosed by Tyler, with the second value, as recited by the independent claims.

See, Advisory at page 3. Thus, in order for the Examiner's contentions to hold true, Tyler must teach or suggest that the polling result (*i.e.,* the second value) is generated by the program (*i.e.,* second thread). However, as discussed above, Tyler actually discloses the polling result is generated by the X-server, ***not*** by the program. This effectively contradicts requirement (*iii*) of the independent claims. Accordingly, Tyler does not satisfy requirement (*iii*) of the independent claims.

In addition, in order for the Examiner's contentions to hold true, it is necessary for the polling result (*i.e.*, second value) to be written using the *stdout* stream of the program (*i.e.,* output stream of the second thread). As discussed above, Tyler discloses the polling result is generated by the X-server and written to the subroutine executing on the controller. However, Tyler also discloses the X-server is directly connected to the controller (*i.e.,* the *stdout* stream does ***not*** connect the X-server to the controller). Accordingly, because the polling result is written to the controller, Tyler effectively discloses the polling result is written ***without*** using the *stdout* stream of the program. This effectively contradicts requirement (*v*) of the independent claims. Accordingly, Tyler does not satisfy requirement (*v*) of the independent claims.

Carlson teaches communication between threads comprising a program counter, a stack, a state, and a register set. *See,* Carlson at column 4, lines 42-65. However, Carlson fails teach what Tyler lacks. Specifically, Carlson, like Tyler, also fails to teach or suggest creating a second value by a second thread, and then writing the second value to a first thread using the input/output streams of the first thread and the second thread. Thus, Carlson, like Tyler, also cannot satisfy at least requirements (*iii*), (*iv*), and (*v*) of the independent claims.

In view of the above, Tyler and Carlson, whether viewed separately or in combination, do not teach or suggest each and every limitation of independent claims 23 and 30. Accordingly, the Examiner's contentions do not support a rejection of independent claims 23 and 30. Claims 26-29 and 33-40 depend, either directly or indirectly, from claims 23 and 30. Accordingly, the Examiner's contentions and the cited prior art also do not support a rejection of claims 26-29 and 33-40.

VIII. Conclusion

In view of the above, the Examiner's contentions and the cited prior art do not support the rejection of claims 23, 26-30, and 33-40 under 35 U.S.C. § 103(a). Accordingly, a favorable decision from the Board is respectfully requested.

Please apply any charges not covered, or any credits, to Deposit Account 50-0591 (Reference Number 36159/098001; P5944).

Dated: October 20, 2008

Respectfully submitted,

By /Robert P. Lord/

Robert P. Lord
Registration No.: 46,479
OSHA · LIANG LLP
3945 Freedom Circle, Suite 300
Santa Clara, California 95054
(408) 727-0600
(713) 228-8778 (Fax)
Attorney for Applicant

Enclosures (Claims Appendix)
(Evidence Appendix)
(Related Proceedings Appendix)

CLAIMS APPENDIX

Claims Involved in Appeal.

23. A method for communicating between threads, comprising:

invoking a first thread;

associating a first input stream and a first output stream with the first thread;

invoking a second thread;

associating a second input stream and a second output stream with the second thread;

invoking a stream operator to write a first data value from the first thread to the second thread, wherein the stream operator connects the first output stream to the second input stream and sends the first data value from the first output stream to the second input stream;

using the second thread to generate a second data value by performing an operation on the first data value; and

invoking the stream operator to write the second data value from the second thread to the first thread, wherein the stream operator connects the second output stream to the first input stream and sends the second data value from the second output stream to the first input stream,

wherein at least one selected from the group consisting of the first thread and the second thread manages an operating system process and comprises:

a program counter;

a stack;

a state; and

a register set.

26. The method of claim 23, wherein the second thread is a child thread of the first thread.

27. The method of claim 23, wherein at least one selected from the group consisting of the first input stream, the first output stream, the second input stream, and the second output stream is a standard stream.

28. The method of claim 27, wherein the standard stream is directly built into a dynamically typed programming language.

29. The method of claim 23, further comprising:

associating a first error stream with the first thread.

30. A computer readable medium storing instructions for communicating between threads, the instructions comprising functionality to:

invoke a first thread;

associate a first input stream and a first output stream with the first thread;

invoke a second thread;

associate a second input stream and a second output stream with the second thread;

invoke a stream operator to write a first data value from the first thread to the second thread, wherein the stream operator connects the first output stream to the second input stream and wherein the stream operator sends the first data value from the first output stream to the second input stream;

use the second thread to generate a second data value by performing an operation on the first data value; and

invoke the stream operator to write the second data value from the second thread to the first thread, wherein the stream operator connects the second output stream to the first input stream and writes the data value from the second output stream to the first input stream,

wherein at least one selected from the group consisting of the first thread and the second thread manages an operating system process and comprises:

a program counter;

a stack;

a state; and

a register set.

33. The computer readable medium of claim 30, wherein the second thread is a child thread of the first thread.

34. The computer readable medium of claim 30, wherein at least one selected from the group consisting of the first input stream, the first output stream, the second input stream, and the second output stream is a standard stream.

35. The computer readable medium of claim 34, wherein the standard stream is directly built into a dynamically typed programming language.

36. The computer readable medium of claim 30, wherein instructions further comprising functionality to:

associate a first error stream with the first thread.

37. The method of claim 23, wherein the first thread and the second thread are associated with a single process.

38. The computer readable medium of claim 30, wherein the first thread and the second thread are associated with a single process.

39. The computer readable medium of claim 30, wherein the stream operator is a symbol.

40. The method of claim 23, wherein the stream operator is a symbol.

EVIDENCE APPENDIX

NONE

RELATED PROCEEDINGS APPENDIX

NONE